Access Control, Authentication, and Authorization

THE FOLLOWING COMPTIA SECURITY+ EXAM OBJECTIVES ARE COVERED IN THIS CHAPTER:

✓ 1.2 Given a scenario, use secure network administration principles.
  - Rule-based management
  - Firewall rules
  - VLAN management
  - Secure router configuration
  - Access control lists
  - Port Security
  - 802.1x
  - Flood guards
  - Loop protection
  - Implicit deny
  - Network separation
  - Log analysis

✓ 1.3 Explain network design elements and components.
  - Layered Security / Defense in Depth

✓ 5.1 Compare and contrast the function and purpose of authentication services.
  - RADIUS
  - TACACS+
- Kerberos
- LDAP
- XTACACS
- SAML
- Secure LDAP

✓ 5.2 Given a scenario, select the appropriate authentication, authorization, or access control.
- Identification vs. authentication vs. authorization
- Authorization: Least privilege; Separation of duties; ACLs; Mandatory access; Discretionary access; Rule-based access control; Role-based access control; Time of day restrictions
- Authentication: Tokens; Common access card; Smart card; Multifactor authentication; TOTP; HOTP; CHAP; PAP; Single sign-on; Access control; Implicit deny; Trusted OS
- Authentication factors: Something you are; Something you have; Something you know; Somewhere you are; Something you do
- Identification: Biometrics; Personal identification verification card; Username
- Federation
- Transitive trust/authentication

✓ 5.3 Install and configure security controls when performing account management, based on best practices.
- Mitigate issues associated with users with multiple account/roles and/or shared accounts
- Account policy enforcement: Credential management; Group policy; Password complexity; Expiration; Recovery; Disablement; Lockout; Password history; Password reuse; Password length; Generic account prohibition
- Group based privileges
- User assigned privileges
- User access reviews
- Continuous monitoring
This chapter covers a critical topic in security, that of controlling who can access your system, what resources they can access, and how to ensure that individuals are who they claim to be. At the most basic level, you can consider authentication and access control to be the two foundations of security. If you don’t do a good job on these tasks, it is unlikely that the rest of your security strategy will be effective.

This chapter starts by looking at the basics of access control and then explores remote access and authentication services. It concludes by examining access control implementation and best practices.

Understanding Access Control Basics

Quite simply, access control means allowing the correct users into a system (those who are authorized) and keeping others out (those who are not authorized). You can employ a great many tools and technologies to make this happen, all of which are discussed in this chapter. The fundamental principle, however, remains the same: Let the right ones in.

In the following sections, we will explore the differences between identification and authentication, authentication and authorization, multifactor authentication, and operational security. We will also look at tokens and issues to watch for.

Identification vs. Authentication

Understanding the difference between identification and authentication is critical to correctly answering access control questions on the Security+ exam. Identification means finding out who someone is. Authentication is a mechanism of verifying that identification. Put another way, identification is claiming an identity; authentication is proving it.

In the physical world, the best analogy would be that any person can claim to be anyone (identification). To prove it (authentication), however, that person needs to provide some evidence, such as a driver’s license, passport, and so forth.

Authentication systems or methods are based on one or more of these five factors:

- Something you know, such as a password or PIN
- Something you have, such as a smart card, token, or identification device
- Something you are, such as your fingerprints or retinal pattern (often called biometrics)
- Something you do, such as an action you must take to complete authentication
- Somewhere you are (this is based on geolocation)
Because of the use of mobile computing, “somewhere you are” authentication is not often used, since users are likely to log in from diverse locations. In fact, many sources do not include “somewhere you are” as an authentication factor.

Systems authenticate each other using similar methods. Frequently, systems pass private information between each other to establish identity. Once authentication has occurred, two systems can communicate in the manner specified in the design. Several common methods are used for authentication, and they fall within the categories of either single factor or multifactor. Each offers something in terms of security and should be considered when you’re evaluating authentication schemes or methods.

Another method that is becoming popular is out-of-band authentication. This is a process whereby the system you are authenticating gets information from public records and asks you questions to help authenticate you. For example, the system might retrieve your credit report and then query you about specific entries in it.

**Authentication (Single Factor) and Authorization**

The most basic form of authentication is known as single-factor authentication (SFA), because only one type of authentication is checked. SFA is most often implemented as the traditional username/password combination. A **username** and **password** are unique identifiers for a logon process. Here’s a synopsis for how SFA works: When users sit down in front of a computer system, the first thing a security system requires is that they establish who they are. Identification is typically confirmed through a logon process. Most operating systems use a **user ID** (username) and **password** to accomplish this. These values can be sent across the connection as plain text or they can be encrypted.

The logon process identifies that you are who you say you are to the operating system and possibly the network. Figure 4.1 illustrates the logon and password process. Notice that the operating system compares this information to the stored information from the security processor, and it either accepts or denies the logon attempt. The operating system might establish privileges or permissions based on stored data about that particular user ID.

Whenever two or more parties authenticate each other, it is known as **mutual authentication**. A client may authenticate to a server and a server may authenticate to a client when there is a need to establish a secure session between the two and employ encryption. Mutual authentication ensures that the client is not unwittingly connecting and providing its credentials to a rogue server, which can then turn around and steal the data from the real server.

Commonly, mutual authentication will be implemented when the data to be sent during the session is of a critical nature, such as financial or medical records.
Multifactor Authentication

When two or more access methods are included as part of the authentication process, you’re implementing a multifactor authentication system. A system that uses smart cards and passwords is referred to as a two-factor authentication system. Two-factor authentication is shown in Figure 4.2. This example requires both a smart card and a logon password process.

A multifactor system can consist of a two-factor system, three-factor system, and so on. As long as more than one factor is involved in the authentication process, it is considered a multifactor system.

For obvious reasons, the two or more factors employed should not be from the same category. Although you do increase difficulty in gaining system access by requiring the user to enter two sets of username/password combinations, it is much preferred to pair a single username/password combination with a biometric identifier or other security check.

When taking the Security+ exam, keep in mind the number of authentication factors in each type. For example, using a smart card and a password is two-factor authentication. However, using a password and a PIN is one-factor authentication because both involve “something you know.”

Layered Security and Defense in Depth

Two terms synonymous with each other are layered security and defense in depth. All these terms mean is that you should not rely on a single entity for protection but instead implement multiple layers of security. In a physical environment, for example, it is all well and good to have a guard posted at the entrance of the office building, but to keep the servers secure, you should also put a lock on the server room door. From a technology standpoint,
a firewall is a great thing to restrict traffic into the network from the outside, but you will also want to have antivirus software, intrusion detection, and as many other layers of security as you can to truly protect the systems.  

**FIGURE 4.2** Two-factor authentication

![Two-factor authentication diagram]

Both factors must be valid:
- User ID and Password
- Smart Card

---

**Network Access Control**

*Operational security* focuses on how an organization achieves its goals. It is also part of a security triad that also includes physical and management security. As such, operational security issues include network access control (NAC), authentication, and security topologies after the network installation is complete. Issues include the daily operations of the network, connections to other networks, backup plans, and recovery plans. In short, operational security encompasses everything that isn't related to design or physical security in your network. Instead of focusing on the physical components where the data is stored, such as the server, the focus is now on the topology and connections.

Some vendors use the acronym NAC to signify *network admission control* rather than the more commonly accepted *network access control*. Regardless of which word appears mid-acronym, the concept is the same.

The issues you address in an operational capacity may seem overwhelming at first. Many of the areas on which you’ll focus are vulnerabilities in the systems you use or weak or inadequate security policies. For example, if you implement a comprehensive password expiration policy, you can require that users change their passwords every 30 or 60 days. If the system
doesn’t require password rotation, though (it allows the same passwords to be reused), you have a vulnerability that you may not be able to eliminate. A user can go through the motions of changing their password only to reenter the same value and keep it in use.

From an operational perspective, this type of system has weak password-changing capabilities. There is nothing you can do short of installing a higher-security logon process or replacing the operating system. Either solution may not be feasible given the costs, conversion times, and possible unwillingness of an organization—or its partners—to make this switch.

Such dependence on a weak system usually stems from the fact that most companies use software that was developed by third parties in order to save costs or meet compatibility requirements. These packages may require the use of a specific operating system. If that operating system has significant security problems or vulnerabilities, your duties will be immense as you’ll still be responsible for providing security in such an environment. Your secure corporate network, for example, should never be connected to the Internet, where it can become subject to a seemingly endless number of potential vulnerabilities. You must install hardware and software solutions to improve security, and you must convince management that these measures are worth the cost to implement.

**Tokens**

*Security tokens* are similar to certificates in that they are used to identify and authenticate the user. They contain the rights and access privileges of the token bearer as part of the token. Think of a token as a small piece of data that holds a sliver of information about the user.

Many operating systems generate a token that is applied to every action taken on the computer system. If your token doesn’t grant you access to certain information, then either that information will not be displayed or your access will be denied. The authentication system creates a token every time a user connects or when a session begins. At the completion of a session, the token is destroyed. Figure 4.3 shows the security token process.

**Federations**

A *federation* is a collection of computer networks that agree on standards of operation, such as security standards. Normally, these are networks that are related in some way. In some cases, it could be an industry association that establishes such standards.

Another example of a federation would be an instant messaging (IM) federation. In this scenario, multiple IM providers form common communication standards, thus allowing users on different platforms with different clients to communicate freely.

In other situations, a group of partners might elect to establish common security and communication standards, thus forming a federation. This would facilitate communication between employees in each of the various partners.

A *federated identity* is a means of linking a user’s identity with their privileges in a manner that can be used across business boundaries (for example, Microsoft Passport or Google checkout). This allows a user to have a single identity that they can use across different business units and perhaps even entirely different businesses.
**FIGURE 4.3** Security token authentication

A federated identity sounds similar to a single sign-on, but do not confuse the two. Single sign-on is about having one password for all resources on a given network. Federated identities relate to being able to access resources on diverse networks.

**Potential Authentication and Access Problems**

There are two problem areas you should know about for the Security+ exam as they apply to authentication/access issues: transitive access and client-side attacks. Let’s address both of these.

**Transitive Access**

The word *transitive* means involving transition—keep this in mind as you learn how *transitive access* problems occur. With transitive access, one party (A) trusts another party (B). If the second party (B) trusts another party (C), then a relationship can exist where the first party (A) also may trust the third party (C).

In early operating systems, this process was often exploited. In current operating systems, such as Windows Server 2012, the problems with transitive access are solved by creating *transitive trusts*, which are a type of relationship that can exist between domains (the opposite is *nontransitive trusts*). When the trust relationship is transitive, the relationship between party (A) and party (B) flows through as described earlier (for instance, A now trusts C). In all versions of Active Directory, the default is that all domains in a forest trust each other with two-way, transitive trust relationships.

Although this process makes administration much easier when you add a new child domain (no administrative intervention is required to establish the trusts), it leaves open
the possibility of a hacker acquiring more trust than they should by virtue of joining the domain. In Exercise 4.1, we’ll explore how to validate the trust relationship in Windows Server 2012, which is a step toward addressing this problem.

**EXERCISE 4.1**

**Validating a Trust Relationship**

As an administrator, you should know what trust relationships exist between domains. To validate a trust relationship in Windows Server 2012, follow these steps:

1. Open Active Directory Domains and Trusts.
2. Right-click your domain name, and choose Properties from the menu.
3. Click the Trusts tab, and select the name of the domain, or forest, that you want to validate.
4. Click Properties. The Properties dialog box for that trust appears.
5. Approximately two-thirds of the way down the dialog box, the Transitivity Of Trust item appears. Click Validate.
6. A confirmation message appears. Click OK.

**Authentication Issues to Consider**

You can set up many different parameters and standards to force the people in your organization to conform. In establishing these parameters, it’s important that you consider the capabilities of the people who will be working with these policies. If you’re working in an environment where people aren’t computer savvy, you may spend a lot of time helping them remember and recover passwords. Each organization has its own quirks, and many have had to reevaluate their security guidelines only after they’ve already invested a great deal of time and expense to implement high-security systems to accommodate such quirks. Remember that it is always better to educate users and raise their awareness than to lower security.

Setting authentication security, especially when supporting users, can become a high-maintenance activity for network administrators. On one hand, you want people to be able to authenticate themselves easily; on the other hand, you want to establish security that protects your company’s resources. Here are some tips for making this process easier:

- Be wary of popular names or current trends that make certain passwords predictable. For example, every January, Super Bowl teams become likely passwords, as do variations on players’ names and numbers. This can create a security problem for computer centers.
- Use *identity proofing* whenever an issue arises between identification and authentication. The identification process starts when a user ID or logon name is typed into
a sign-on screen. Authentication is accomplished by challenging the claim of who is accessing the resource. In other words, you start by claiming to be some specific user (perhaps an administrator); then authentication asks you to prove it (perhaps by providing the proper password).

- Incorporate a second value, such as mother’s maiden name, to prove a user’s identity. This is helpful when identity proofing is invoked—for example, in the case of a lost password and a person claims that they are a given user but cannot be authenticated.

---

### Real World Scenario

**Multifactor Authentication and Security**

The IT manager of your company is becoming increasingly concerned about the laxness of users when it comes to computer security. She reports that users regularly leave the office at the end of the day without signing out of their accounts.

The company is attempting to win a contract that involves government work, which will require additional security measures. What would you suggest?

First and foremost, you should recommend that the company implement a multifactor authentication system. This system could consist of a smart card and a logon/password process. Most smart card readers can be configured to require that the card remain inserted in the reader while the user is logged on. If the smart card is removed, say at the end of the day, the workstation will automatically log the user out. By requiring a logon/password process, you can still provide security if the smart card is stolen. This solution provides a reasonable level of security, and it doesn’t significantly increase security costs. It works best if the smart card is also required to exit the building, otherwise you run the risk of people leaving their cards at their desk when they go home.

Other suggestions are to consider additional access controls, such as perimeter alarms and physical access control of sensitive areas. The government would probably require these anyway, although such measures won’t force users to log out when they leave their workstations.

An inherent problem with many identity-proofing implementations is that they ask questions that someone other than the user could easily guess or learn (for example, what color are your eyes?). To increase the difficulty of someone fraudulently using identity proofing, you should use only questions that are difficult to guess, or implement biometrics such as voice identification. Under no circumstances should the person proofing be allowed access immediately; instead, their access information should be sent to their email account of record.
Authentication Protocols

A variety of authentication protocols are used to aid in authenticating a user (or another system) to a system:

- **PAP (Password Authentication Protocol)** is an older system that is no longer used. PAP sends the username and password to the authentication server in plain text.
- **SPAP (Shiva Password Authentication Protocol)** replaced PAP. The main difference is that SPAP encrypts the username and password.
- **CHAP (Challenge Handshake Authentication Protocol)** was designed to stop man-in-the-middle attacks. During the initial authentication, the connecting machine is asked to generate a random number (usually a hash) and send it to the server. Periodically the server will challenge the client machine, demanding to see that number again. If an attacker has taken over the session, they won’t know that number and won’t be able to authenticate.
- The **TOTP (Time-Based One-Time Password)** algorithm uses a time-based factor to create unique passwords.
- The **HOTP (HMAC-Based One-Time Password)** algorithm is based on using a Hash Message Authentication Code (HMAC) algorithm. HMAC will be discussed in Chapter 7, “Host, Data, and Application Security.”

Account Policy Enforcement

The account policy determines the security parameters regarding who can and cannot access the system. As mentioned earlier, there is a fine line between lax security (which keeps users happy) and stringent security. When you impose stringent security—long passwords that must be changed every few days and accounts locked as soon as wrong entries are given—you create unhappy users who will often start jeopardizing the very security you are trying to create by writing down values on slips of paper that can fall in the wrong hands.

In this section, we will look at the best practices related to key components of account policy enforcement that you need to know for the exam. These include the issues of password length and complexity, password expiration, password recovery, and, finally, password disablement and lockout.

Password Length and Complexity

The more difficult a user’s password is, the more difficult it becomes for a miscreant to break it and log in as that user, and the more difficult it becomes, as well, for the user to remember it. Thus, you need to obtain a fine balance between the two extremes.

Eight characters (upper and lowercase) are generally considered the minimum for password length, and most systems today encourage the use of at least one non-alpha character—punctuation, special characters, numbers, and so on. On Windows-based systems, the password value can be set to 0 to not require passwords.
Windows 8 (as well as older Windows versions such as Windows 7 and Vista)—through the Local Security Policy (which is overridden by Group Policy values on a domain controller)—allows you to choose to enable password complexity. Choosing this option requires users to create passwords that meet the following requirements:

- They cannot contain the user’s account name or parts of the user’s full name that exceed two consecutive characters.
- They must be at least eight characters long.
- They have to contain characters from at least three of the following four sets:
  - A–Z
  - a–z
  - 0–9
  - Non-alpha characters (!, $, #, %, and so forth)

There are methods that will allow a person with physical access to a Windows machine to retrieve the password regardless of complexity. You can find tools such as Ophcrack freely available on the Web for this purpose. For this reason, many experts recommend longer passphrases with complexity. For example, !L!k3Ch33s3Burg3rsFromBurg3rK!n3g would be something the user can remember but would be difficult to guess or crack.

**Password Expiration**

Every password must expire because the longer the same value is used, the more likely it is to be broken. Ninety days is acceptable for many organizations, but Microsoft often recommends setting this value to 42 days if you want to enforce strong password usage throughout the organization. For more information on this topic, go to http://technet.microsoft.com/en-us/library/cc875814.aspx.

To keep users from changing their password to the same value as the old one, or to one they used the last time around, you should enable password history. Most Microsoft OSs allow you to set this to a number between 0 (disabled) and 24. For the best security, set it to 24 so that 24 unique passwords must be used by any given user before they can begin to reuse them.

Along with the expiration date, you should configure a minimum number of days that can exist between password changes. If this setting is disabled or set too low, users can immediately change new passwords to other values—something a hacker may want to do. We recommended that you not set it to anything lower than 2 days.

**Password Recovery**

One of the certainties in life is that users will occasionally forget their password. This often occurs shortly after they’ve changed it from one value to another, but it can often occur after a long weekend. Since a user’s password isn’t stored on most operating systems (only a
hash value is kept), most operating systems allow the administrator to change the value for a user who has forgotten theirs. This new value allows the user to log in and then immediately change it to another value that they can (ideally) remember.

**Password Disablement and Lockout**

When a user will be gone from a company for a while (maternity leave, for example), their account should be disabled until they return. When a user will be gone from a company forever (termination), their account should be removed from the system immediately.

If there is the possibility that the terminated employee may come back as a contractor, then consider suspending the account as opposed to removing it. This action is preferable since relative identifiers (RIDs) are not reused after they are deleted.

Between the two extremes lies the need to lock an account. This occurs when a user is attempting to log in but giving incorrect values; locking this account is necessary to prevent a would-be attacker from repeatedly guessing at password values until they find a match. You can configure the lockout policies at the local level on the workstation (Local Security Policy) as well as at the domain level (Group Policies), and the values you configure are the same:

**Account Lockout Duration**  When the system locks the account, this is the duration before it is unlocked. With Windows, this value can range from 0 minutes to 99,999 minutes. Setting it to 0 does not disable the feature but rather requires an administrator to explicitly unlock the account before it can be used again.

**Account Lockout Threshold**  This setting determines how many incorrect attempts a user can give before the account is locked. In Windows, this value can range from 0 to 999 failed attempts. If it is set at 0, the account will never be locked out. Note that attempts to enter values at the password-protected screensaver count just the same as attempts to log in after the system has booted and Ctrl+Alt+Delete has been pressed.

**Reset Account Lockout Counter After**  This value specifies the number of minutes to wait between counting failed login attempts that are part of the same batch of attempts. For example, Account Lockout Threshold may be set to 3 to lock the account after three bad tries, but this value can be set to 5 so that if the user tries once and then waits more than five minutes to try again, they have another three attempts before the account is locked. The values here can range from 0 to 9,999 minutes. This value can be set (or have meaning) only if the Account Lockout Threshold is set, and the reset time must be less than or equal to the Account Lockout Duration value.

**Users with Multiple Accounts/Roles**

As we mentioned when discussing least privilege, of particular concern is users who have multiple accounts and/or act in multiple roles. An example of this is any administrator who
has an account they use for administrative purposes and one they use when performing another role (editor, author, etc.). In most organizations, it is not possible to not have a number of users who operate in multiple capacities. The key then becomes education and policies.

Education is needed to show why employees should use the elevated accounts only when necessary and to make them understand the security risks inherent in operating at those levels. Policies are needed to put into action and enforce the common sense that these users should possess. The policy should be understood—and signed off on—by those operating in this group.

**Generic Account Prohibition**

A generic account is any account that is shared—allowing multiple users to log in and use the system/network/resource. Not only can these be guest accounts, but they can also be anonymous accounts, accounts created for temporary users, and a plethora of other possibilities. While such generic accounts can make it easy to grant access to the system, they suffer from two significant problems. The first is the password is shared, and that goes against normal security procedures. The second significant problem is that since the account is being shared, auditing it can be a nightmare. As a general rule, therefore, generic accounts should not be used where they can be avoided.

**Group-based and User-assigned Privileges**

Two methods of privilege assignment prevalent today are group-based and user-assigned. As the name implies, group-based privileges are those acquired as a result of belonging to a group. This topic is touched on in several locations throughout this chapter (with the discussion of group policy, and role-based access control), but the easiest way to think of it is that if you are member of the editors’ group, then you have access to resources needed by editors and it is easier for the administrator to grant permissions to all members of that group than to individually assign them to each.

User-assigned privileges are those that can be assigned by the user. For example, when you create a file in most operating systems, you can change the permissions associated with that file. It is possible, for example, to give others the privilege of only being able to read it, or to read and write to it.

**Understanding Remote Access Connectivity**

One of the primary purposes of having a network is the ability to connect systems. As networks have grown, many technologies have come on the scene to make this process easier and more secure. A key area of concern relates to the connection of systems and other networks that aren’t part of your network. The following sections discuss the more common protocols used to facilitate connectivity among remote systems.
Ancient History: The Serial Line Internet Protocol

*Serial Line Internet Protocol (SLIP)* is an older protocol that was used in early remote access environments and serves as the starting point for most remote discussions. SLIP was originally designed to connect Unix systems in a dial-up environment, and it only supported serial communications.

A very simple protocol, SLIP could only be used to pass TCP/IP traffic and wasn’t secure or efficient. Although some systems today still support SLIP, it is strictly there for legacy systems and should be avoided whenever possible.

Any authentication done for a remote user is known as *remote authentication*. This authentication is commonly done using TACACS, TACACS+, XTACACS, or RADIUS.

Using the Point-to-Point Protocol

Introduced in 1994, the *Point-to-Point Protocol (PPP)* offers support for multiple protocols, including AppleTalk, IPX, and DECnet. PPP works with POTS, Integrated Services Digital Network (ISDN), and other faster connections, such as T1. PPP doesn’t provide data security, but it does provide authentication using the *Challenge Handshake Authentication Protocol (CHAP)*.

Except for CHAP, the aforementioned protocols are just given for background information and are not on the Security+ exam. CHAP is described in more detail elsewhere in this chapter.

Figure 4.4 shows a PPP connection over an ISDN line. In the case of ISDN, PPP would normally use one 64 Kbps B channel for transmission. PPP allows many channels in a network connection (such as ISDN) to be connected or bonded together to form a single virtual connection.

PPP works by encapsulating the network traffic in a protocol called the *Network Control Protocol (NCP)*. Authentication is handled by the *Link Control Protocol (LCP)*. A PPP connection allows remote users to log on to the network and have access as though they were local users on the network. PPP doesn’t provide for any encryption services for the channel.

As you may have guessed, the unsecure nature of PPP makes it largely unsuitable for WAN connections. To counter this issue, other protocols have been created that take advantage of PPP’s flexibility and build on it. You should make sure that all of your PPP connections use secure channels, dedicated connections, or high-speed connections.

Remote users who connect directly to a system don’t necessarily need to have encryption capabilities enabled. If the connection is direct, the likelihood that anyone would be able to tap an existing phone line is relatively small. However, you should make sure that connections through a network use an encryption-oriented tunneling system.
Working with Tunneling Protocols

Tunneling protocols add the ability to create tunnels between networks that can be more secure, support additional protocols, and provide virtual paths between systems. The best way to think of tunneling is to imagine sensitive data being encapsulated in other packets that are sent across the public network. Once they’re received at the other end, the sensitive data is stripped from the other packets and recompiled into its original form.

The most common protocols used for tunneling are as follows:

**Point-to-Point Tunneling Protocol**  Point-to-Point Tunneling Protocol (PPTP) supports encapsulation in a single point-to-point environment. PPTP encapsulates and encrypts PPP packets. This makes PPTP a favorite low-end protocol for networks. The negotiation between the two ends of a PPTP connection is done in the clear. After the negotiation is performed, the channel is encrypted. This is one of the major weaknesses of PPTP. A packet-capture device, such as a sniffer, that captures the negotiation process can potentially use that information to determine the connection type and information about how the tunnel works. Microsoft developed PPTP and supports it on most of its products. PPTP uses port 1723 and TCP for connections.

**Layer 2 Forwarding**  Layer 2 Forwarding (L2F) was created by Cisco as a method of creating tunnels primarily for dial-up connections. It’s similar in capability to PPP and shouldn’t be used over WANs. L2F provides authentication, but it doesn’t provide encryption. L2F uses port 1701 and TCP for connections.

**Layer 2 Tunneling Protocol**  Microsoft and Cisco agreed to combine their respective tunneling protocols into one: Layer 2 Tunneling Protocol (L2TP). L2TP is a hybrid of PPTP and L2F. It’s primarily a point-to-point protocol. L2TP supports multiple network protocols and can be used in networks beside TCP/IP. L2TP works over IPX, SNA, and IP, so it can be used as a bridge across many types of systems. The major problem with L2TP is that it doesn’t provide data security—the information isn’t encrypted. Security can be provided by protocols such as IPSec. L2TP uses port 1701 and UDP for connections.
Secure Shell  Secure Shell (SSH) is a tunneling protocol originally designed for Unix systems. It uses encryption to establish a secure connection between two systems. SSH also provides alternative, security-equivalent programs for such Unix standards as Telnet, FTP, and many other communications-oriented applications. SSH is available for use on Windows systems as well. This makes it the preferred method of security for Telnet and other cleartext-oriented programs in the Unix environment. SSH uses port 22 and TCP for connections.

Internet Protocol Security  Internet Protocol Security (IPSec) isn’t a tunneling protocol, but it’s used in conjunction with tunneling protocols. IPSec is oriented primarily toward LAN-to-LAN connections, but it can also be used with remote connections. IPSec provides secure authentication and encryption of data and headers, which makes it a good choice for security. IPSec can work in either Tunneling mode or Transport mode. In Tunneling mode, the data or payload and message headers are encrypted. Transport mode encrypts only the payload. IPSec is an add-on to IPv4 and built into IPv6.

Working with RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a mechanism that allows authentication of remote and other network connections. Originally intended for use on dial-up connections, it has moved well beyond that and offers many state-of-the-art features. The RADIUS protocol is an IETF standard, and it has been implemented by most of the major operating system manufacturers. A RADIUS server can be managed centrally, and the servers that allow access to a network can verify with a RADIUS server whether an incoming caller is authorized. In a large network with many connections, this allows a single server to perform all authentications.

The term caller may seem outdated, but Windows Server 2012 (and 2008 as well as 2003) all refer to the ability to remotely access a system as dial-in privileges. Although few people are actually “dialing” or calling in, the terms have stuck.

Figure 4.5 shows an example of a RADIUS server communicating with an ISP to allow access to a remote user. Notice that the remote ISP server is functioning as a client to the RADIUS server. This allows centralized administration of access rights.

You should use RADIUS when you want to improve network security by implementing a single service to authenticate users who connect remotely to the network. Doing so gives you a single source for the authentication to take place. Additionally, you can implement auditing and accounting on the RADIUS server.

The major difficulty with a single-server RADIUS environment is that the entire network may refuse connections if the server malfunctions. Many RADIUS systems allow multiple servers to be used to increase reliability. All of these servers are critical components of the infrastructure, and they must be protected from attack.
FIGURE 4.5 The RADIUS client manages the local connection and authenticates against a central server.

TACACS/TACACS+/XTACACS

*Terminal Access Controller Access-Control System (TACACS)* is a client/server-oriented environment, and it operates in a manner similar to RADIUS. Extended TACACS (XTACACS) replaced the original version and combined authentication and authorization with logging to enable auditing.

The most current method, or level, of TACACS is TACACS+. It replaces the previous two incarnations. TACACS+ allows credentials to be accepted from multiple methods, including Kerberos. The TACACS client/server process occurs in the same manner as the RADIUS process illustrated in Figure 4.5.

Cisco has widely implemented TACACS+ for connections. TACACS+ has become a widely accepted as an alternative to RADIUS.

Remember, RADIUS and TACACS (or any of its variations such as TACACS+ or XTACACS) can be used to authenticate connections.

VLAN Management

*A virtual local area network (VLAN)* allows you to create groups of users and systems and segment them on the network. This segmentation lets you hide segments of the network from other segments and thereby control access. You can also set up VLANs to control the paths that data takes to get from one point to another. A VLAN is a good way to contain network traffic to a certain area in a network.

Think of a VLAN as a network of hosts that act as if they’re connected by a physical wire even though there is no such wire between them.
On a LAN, hosts can communicate with each other through broadcasts, and no forwarding devices, such as routers, are needed. As the LAN grows, so too does the number of broadcasts. Shrinking the size of the LAN by segmenting it into smaller groups (VLANs) reduces the size of the broadcast domains. The advantages of doing this include reducing the scope of the broadcasts, improving performance and manageability, and decreasing dependence on the physical topology. From the standpoint of this exam, however, the key benefit is that VLANs can increase security by allowing users with similar data sensitivity levels to be segmented together.

**SAML**

*Security Assertion Markup Language (SAML)* is an open standard based on XML that is used for authentication and authorization data. Service providers often use SAML to prove the identity of someone connecting to the service provider. The current version is SAML v2.0.

**Understanding Authentication Services**

Authentication services are the implementation of the specific technology in question. For this part of the exam, the focus is on LDAP and Kerberos, though many other possibilities exist, such as Internet Authentication Service (IAS) and Central Authentication Service (CAS), which are outside the scope of this exam. Single sign-on initiatives round out the discussion in this section.

**LDAP**

*Lightweight Directory Access Protocol (LDAP)* is a standardized directory access protocol that allows queries to be made of directories (specifically, pared-down X.500-based directories). If a directory service supports LDAP, you can query that directory with an LDAP client, but it’s LDAP itself that is growing in popularity and is being used extensively in online white and yellow pages.

LDAP is the main access protocol used by Active Directory. It operates, by default, at port 389. The LDAP syntax uses commas between names.

Because a breach of LDAP can be quite serious, some organizations use secure LDAP. With *secure LDAP* (LDAPS), all LDAP communications are encrypted with SSL/TLS, and port 636 is used.

Throughout this book you will see various port numbers mentioned. These port numbers are often the subject of questions on the Security+ exam (as well as other security-related certifications), so it is a good idea for you to get to know them.
Kerberos

Kerberos is an authentication protocol named after the mythical three-headed dog that stood at the gates of Hades. Originally designed by MIT, Kerberos is very popular as an authentication method. It allows for a single sign-on to a distributed network.

Kerberos authentication uses a key distribution center (KDC) to orchestrate the process. The KDC authenticates the principal (which can be a user, program, or system) and provides it with a ticket. After this ticket is issued, it can be used to authenticate against other principals. This process occurs automatically when another principal performs a request or service.

Kerberos is quickly becoming a common standard in network environments. Its only significant weakness is that the KDC can be a single point of failure. If the KDC goes down, the authentication process will stop. Figure 4.6 illustrates the Kerberos authentication process and the ticket being presented to systems that are authorized by the KDC.

When using Kerberos, the user authenticates to the KDC and is given a ticket granting ticket (TGT). This ticket is encrypted and has a time limit of up to 10 hours. The ticket lists the privileges of that user (much like a token). Each time the user wishes to access some resource on the network, the user’s computer presents the KDC with the TGT; the TGT then sends that user’s computer a service ticket, granting the user access to that service. Service tickets are usually only good for up to 5 minutes. The user’s computer then sends the service ticket to the server the user is trying to access. As a final authentication check, that server then communicates with the TGT to confirm and validate the service ticket.

**Figure 4.6** Kerberos authentication process

1. User requests access to service running on a different server.
2. KDC authenticates user and sends a ticket to be used between the user and the service on the server.
3. User’s workstation sends a ticket to the service.
Single Sign-On Initiatives

One of the big problems that larger systems must deal with is the need for users to access multiple systems or applications. This may require a user to remember multiple accounts and passwords. The purpose of a single sign-on (SSO) is to give users access to all the applications and systems they need when they log on. This is becoming a reality in many environments, including Kerberos, Microsoft Active Directory, Novell eDirectory, and some certificate model implementations.

**NOTE**

Single sign-on is both a blessing and a curse. It’s a blessing in that once the user is authenticated, they can access all of the resources on the network and browse multiple directories. It’s a curse in that it removes the doors that otherwise exist between the user and various resources.

In the case of Kerberos, a single token allows any “Kerberized” applications to accept a user as valid. The important thing to remember in this process is that each application that wants to use SSO must be able to accept and process the token presented by Kerberos.

Active Directory (AD) uses a slightly different method. A server that runs AD retains information about all access rights for all users and groups in the network. When a user logs on to the system, AD issues the user a globally unique identifier (GUID). Applications that support AD can use this GUID to provide access control.

Figure 4.7 illustrates this process in further detail. In this instance, the database application, email client, and printers all authenticate with the same logon. Like Kerberos, this process requires all applications that want to take advantage of AD to accept AD controls and directives.

**FIGURE 4.7** AD validating a user
In this way, the user doesn’t have to have separate sign-on, email, and application passwords. Using AD simplifies the sign-on process for users and lowers the support requirements for administrators. Access can be established through groups, and it can be enforced through group memberships. For example, the domain administrator can place all the tech support people in a group named techsupport and assign privileges to the entire group.

On a decentralized network, SSO passwords are stored on each server and can represent a security risk. It’s important to enforce password changes and make certain that passwords are updated throughout the organization on a frequent basis.

Although single sign-on has its own security issues, it is perhaps the simplest way to mitigate the issue of users with multiple roles and multiple accounts. If a user is required to manage a plethora of accounts/login credentials, the natural tendency is to write down usernames and passwords, which is a major security risk. Single sign-on, if managed properly, can be an excellent answer to this issue.

**Single sign-on is not the opposite of multifactor authentication, but it is often mistakenly thought of this way. One-, two-, and three-factor authentication merely refer to the number of items a user must supply to authenticate. Authentication can be based on something they have (a smart card), something they know (a password), something unique (biometric), and so forth. Once factor authentication is done, single sign-on can still apply throughout a user’s session.**

### Understanding Access Control

The four primary methods of access control are as follows:

- **Mandatory Access Control (MAC)**  
  All access is predefined.

- **Discretionary Access Control (DAC)**  
  Incorporates some flexibility.

- **Role-Based Access Control (RBAC)**  
  Allows the user’s role to dictate access capabilities.

- **Rule-Based Access Control (RBAC)**  
  Rule-Based Access Control (which also uses the RBAC acronym) is gaining in popularity and limits the user to settings in preconfigured policies.

Each of these methods has advantages and disadvantages to the organization from a security perspective—that’s why they all still exist. Each is appropriate for some environment.

There has been some recent discussion of something called **Lattice-Based Control (LBAC)**. LBAC is a variation of Mandatory Access Control, and it isn’t addressed separately on the Security+ exam. It involves a lattice composed of subjects (users, systems, and so forth) and resources, and the resources are labeled to provide access control.

The method you choose will be greatly affected by your organization’s philosophy on the sharing of information. In a high-security environment, the tendency is to implement either
a MAC or RBAC (in this case Role-Based Access Control) method. In a traditional business environment or educational institution, the tendency is to implement a DAC method. You should do some consulting within the organization to understand how particular departments and the organization as a whole want to implement access control models. Doing so will allow you to gather input from all concerned parties about how to establish access guidelines and how to implement security.

In the following sections, we will look at each of these methods from a business perspective.

**Mandatory Access Control**

*Mandatory Access Control (MAC)* is a relatively inflexible method for how information access is permitted. In a MAC environment, all access capabilities are predefined. Users can’t share information unless their rights to share it are established by administrators. Consequently, administrators must make any changes that need to be made to such rights. This process enforces a rigid model of security. However, it is also considered the most secure security model.

For a MAC model to work effectively, administrators and network designers must think relationships through carefully in advance of implementation. The advantage of this model is that security access is well established and well defined, making security breaches easier to investigate and correct. A well-designed MAC model can make the job of information control easier and can essentially lock down a network. The major disadvantages of this model are its lack of flexibility and the fact that it requires change over time. The inability of administrative staff to address these changes can sometimes make the model hard to maintain.

This model is used in environments where confidentiality is a driving force. It often employs government and military classifications (labels), such as Top Secret and others.

**Discretionary Access Control**

In a *Discretionary Access Control (DAC)* model, network users have some flexibility regarding how information is accessed. This model allows users to share information dynamically with other users. The method allows for a more flexible environment, but it increases the risk of unauthorized disclosure of information. Administrators have a more difficult time ensuring that information access is controlled and that only appropriate access is issued.

A classic example of DAC is the permission structure that exists for “other” with files in the Unix/Linux environment. All permissions in this operating system fall within three groups of users: owner, group, and other. The permissions associated with the owner and the group to which the owner belongs are based on their roles, but all of those who are not the owner, or a member of the owner’s group, fall within the category of other.

The permissions for this group are set separately from the other two and, with very few special exceptions, are a combination of read, write, and execute. Within this environment, you can create a database and give yourself (owner) permission to read and write, give other admins (group) only read permission, and not give any permission to those not in admin (other).
You could just as easily create a script file that cleans up log files and frees space on a workstation. To do this, you would give yourself (owner) all rights, give other admins (group) the ability to read and execute, and give basic users (other) the right only to execute.

**Role-Based Access Control**

*Role-Based Access Control (RBAC)* models approach the problem of access control based on established roles in an organization. RBAC models implement access by job function or by responsibility. Each employee has one or more roles that allow access to specific information. If a person moves from one role to another, the access for the previous role will no longer be available. RBAC models provide more flexibility than the MAC model and less flexibility than the DAC model. They do, however, have the advantage of being strictly based on job function as opposed to individual needs.

Instead of thinking “Denise needs to be able to edit files,” RBAC uses the logic “Editors need to be able to edit files” and “Denise is a member of the Editors group.” This model is always good for use in an environment in which there is high employee turnover.

This is also sometimes called *group-based control* or *group-based permissions*. Essentially, Windows operating systems work in this fashion. Your permissions on a Windows-based domain are determined by the group(s) into which you are placed. These groups are, in effect, roles.

**Rule-Based Access Control**

*Rule-Based Access Control (RBAC)* uses the settings in preconfigured security policies to make all decisions. These rules can be to:

- Deny all but those who specifically appear in a list (an allow list)
- Deny only those who specifically appear in the list (a true deny list)

Entries in the list may be usernames, IP addresses, hostnames, or even domains. Rule-Based models are often being used in conjunction with Role-Based to add greater flexibility.

The easiest way to implement Rule-Based Access Control is with access control lists (ACLs), discussed later in this chapter. The ACLs create the rules by which the access control model functions.

**Implementing Access Controlling Best Practices**

How you implement access control makes all the difference in the security of your systems. In this section, we will look at smart cards, access control lists, trusted operating systems, secure router configuration, and a few others.
Least Privileges

This is one of the most critical concepts in access control. Implementing least privileges means that any given user (or system) is given the minimum privileges necessary to accomplish his or her job. For example, if sales managers need to run reports from a database, they will be given privileges only allowing the running of reports. They won’t be given privileges to delete data, alter the database tables, add users, and so forth.

Any privilege could be used to cause some harm to the system, even inadvertently, for example, the deletion of important data. This is particularly true when users have more privileges than they really need. Also, privilege escalation is a common attack on systems. Giving each user the minimum privileges to accomplish their job reduces this risk.

In the real world, you will find some resistance to this idea. Some employees will perceive it as a lack of trust. For example, a software engineer, who may have more experience and more training than the security administrator, might be upset to learn that she does not have unrestricted access to the database. One way to address this is to educate end users. One fact that tends to be compelling to users is this: If they don’t have access to a given system/resource and if something goes awry with that resource, they cannot be held responsible.

Separation of Duties

Almost every operating system in use today employs the concept of differentiation between users and groups at varying levels. As an example, there is always a system administrator (SA) account that has godlike control over everything: root in Unix/Linux, admin (or a deviation of it) in Windows, administrator in Apple OS X, supervisor in Novell NetWare, and so on. Once you move beyond that user, you move to administrative accounts, then regular users, and all the way down to restricted accounts, which can barely do more than log in.

As a security administrator, you need to use that as a baseline and then go beyond that and make certain that you have as many different levels of permissions and privileges as possible. At a minimum, you should do the following:

- Separate the SA account from regular accounts. Never log in as the SA and use the system to perform routine functions. Use the SA account only to do those operations that require those privileges.
- Limit the SA account to as small a group as possible.
- Separate the audit and logging responsibilities from the SA.

Again, using the ISO standard as an example, it recommends the segregation of duties and separation of environments as a way to reduce the likelihood of misuse of systems or information (either intentional or accidental).

Time of Day Restrictions

One of the easiest policies to enforce is time of day restrictions. Almost every operating system—server and workstation—allows you to configure when an account can have access.
to the system. While it may seem pedantic, it can increase the security of the system significantly. For example, if you are the administrator for an office, and workers use the systems only from 8:00 to 5:00 Monday through Friday, then you can configure their accounts to allow access only from 7:00 to 6:00 (offering an extra hour at each end for work they need to do outside of normal) on those days and not allow access outside of those parameters. What you have accomplished by making the accounts valid for only 55 hours each week is to prevent them from being used by attackers the other 113 hours. As simple as it is, it is also effective.

Restrictions can be applied as policies for groups or users in Active Directory, set locally, or set through a number of add-on packages.

**User Access Review**

In addition to assigning user access properly, it is important to review that access periodically. *Access review* is a process to determine whether a user’s access level is still appropriate. People’s roles within an organization can change over time. It is important to review user accounts periodically and determine if they still require the access they currently have. An example of such a scenario would be a network administrator who was responsible for the domain controller but then moved over to administer the remote access servers. The administrator’s access to the domain controller should now be terminated. This concept of access review is closely related to the concept of least privileges. It is important that users do not have “leftover” privileges from previous job roles.

Another closely related topic is continuous monitoring. *Continuous monitoring* implies an ongoing audit of what resources a user actually accesses. This can be critical for stopping insider threats. For example, a human resources employee would need access to employee files. However, if that employee is accessing a given employee’s file without a valid work-related reason, this is a security breach. Only by continuously monitoring access can you detect such breaches.

> It turns out that Edward Snowden was using other users’ accounts to access tremendous amounts of data at the National Security Administration (NSA). It was also learned that the NSA location where Snowden was working lacked continuous monitoring. One has to wonder, if they had such monitoring, would they have detected his anomalous activities and prevented a massive security breach?

**Smart Cards**

*Smart cards* are generally used for access control and security purposes. The card itself usually contains a small amount of memory that can be used to store permissions and access information.
Implementing Access Controlling Best Practices

Smart cards are difficult to counterfeit, but they’re easy to steal. Once a thief has a smart card, they have access to all that the card allows. To prevent this, many organizations don’t put any identifying marks on their smart cards, making it harder for someone to use them. A password or PIN is required to activate most smart cards, and encryption is employed to protect the contents. With many smart cards, if you enter the wrong PIN number multiple times (usually three), the card will shut down to enhance security further.

Many European countries are beginning to use smart cards instead of magnetic-strip credit cards because they offer additional security and can contain more information.

<table>
<thead>
<tr>
<th>Real World Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working with Smart Cards</td>
</tr>
<tr>
<td>You’ve been asked to help troubleshoot a problem that is occurring in your school’s computer lab. Students are complaining about viruses that are infecting the flash drives they bring to school. How can you help remedy this situation?</td>
</tr>
<tr>
<td>Make sure that all of the systems in your school lab computers are running antivirus software and that this software is kept up to date. Doing so will prevent known viruses from entering the school’s system and being transferred to student files.</td>
</tr>
<tr>
<td>You may also want to evaluate whether the school’s computers should have removable media installed on their systems. Several manufacturers sell systems called thin clients, which don’t provide any disk storage or removable media on their workstations. Thin clients access servers to download applications, data, and any other information they need to have in order to run. This eliminates the danger of viruses being introduced from student disks.</td>
</tr>
</tbody>
</table>

There are two main types of smart cards: Common Access Cards and Personal Identification Verification Cards. We will discuss these smart cards in the following sections.

Common Access Card

One type of smart card is the Common Access Card (CAC). These cards are issued by the Department of Defense (DoD) as a general identification/authentication card for military personnel, contractors, and non-DoD employees. A picture appears on the front of the card with an integrated chip beneath and a barcode. On the back of the card is a magnetic strip and another barcode.

A CAC is used for accessing DoD computers, signing email, and implementing PKI. In 2008, the most recent year for which data is available, over 17 million cards had been issued. You can find current information on the CAC here: www.cac.mil.
Personal Identification Verification Card

What the CAC is for military employees, the Personal Identity Verification (PIV) (referred to as Personal Identification Verification Card) is to federal employees and contractors. Per Homeland Security Presidential Directive number 12 (HSPD-12), the PIV will eventually be required of all U.S. government employees and contractors. The PIV will be required to gain access (physical and logical) to government resources.

Access Control Lists

Access control lists (ACLs) enable devices in your network to ignore requests from specified users or systems or to grant them access to certain network capabilities. You may find that a certain IP address is constantly scanning your network, and you can block this IP address at the router and the IP address will automatically be rejected any time it attempts to utilize your network.

ACLs allow a stronger set of access controls to be established in your network. The basic process of ACL control allows the administrator to design and adapt the network to deal with specific security threats.

The following sections look at approaches to ACLs, including implicit deny and firewall rules.

Implicit Deny

Within ACLs exists a condition known as implicit deny. An implicit deny clause is implied at the end of each ACL, and it means that if the proviso in question has not been explicitly granted, then access is denied.

Real World Scenario

The Implicit Deny Bouncer

Suppose you’re hosting a party at your home and you give the guest list to a strong friend of yours. As each guest arrives, your friend examines the list to see if their name appears on it. If their name is not on the list, then the “party crasher” is denied entry (access). You don’t have to tell your friend not to let in specific people; only if those attempting entry are not on the guest list, they are implicitly denied access.

The same principle holds true in the ACL. The entity denied access because it does not appear on a list can be a source address, a destination address, a packet type, or almost anything else for which you want to deny access.
Firewall Rules

Firewall rules act like ACLs, and they are used to dictate what traffic can pass between the firewall and the internal network. Three possible actions can be taken based on the rule’s criteria:

- Block the connection
- Allow the connection
- Allow the connection only if it is secured

The rules can be applied to inbound traffic or outbound traffic and any type of network (LAN, wireless, VPN, or remote access). You should audit the firewall rules on a regular basis, verify that you are obtaining the results you want, and make modifications as necessary.

Port Security

Port security involves the Coast Guard keeping our seaports safe from terrorism. Not really. In the realm of IT, port security works at level 2 of the OSI model and allows an administrator to configure switch ports so that only certain MAC addresses can use the port. This is a common feature on both Cisco’s Catalyst as well as Juniper’s EX Series switches and essentially differentiates so-called dumb switches from managed (or intelligent) switches. Similarly, Dynamic ARP Inspection (DAI) works with these and other smart switches to protect ports from ARP spoofing.

Three areas of port security that CompTIA wants you to be familiar with for the Security+ exam are as follows:

MAC Limiting and Filtering  Limit access to the network to MAC addresses that are known, and filter out those that are not. Even in a home network, you can implement MAC filtering with most routers and typically have an option of choosing to allow only computers with MAC addresses that you list or deny only computers with MAC addresses that you list.

**NOTE**  If you don’t know a workstation’s MAC address, use `ipconfig /all` to find it in the Windows-based world (it is listed as `physical address`) and use `ip a` or `ifconfig` in Unix/Linux.

MAC filtering is not foolproof, and a quick look in a search engine will turn up tools that can be used to change the MAC address and help miscreants circumvent this control.

802.1X  As discussed in the following section, adding port authentication to MAC filtering takes security for the network down to the switch port level and increases your security exponentially.

Unused Ports  All ports not in use should be disabled. Otherwise, they present an open door for an attacker to enter.
Working with 802.1X

The IEEE standard 802.1X defines port-based security for wireless network access control. As such, it offers a means of authentication and defines the Extensible Authentication Protocol (EAP) over IEEE 802—discussed in Chapter 12, “Disaster Recovery and Incident Response”—and is often known as EAP over LAN (EAPOL). The biggest benefit of using 802.1X is that the access points and the switches do not need to do the authentication but instead rely on the authentication server to do the actual work.

Flood Guards and Loop Protection

A flood guard is a protection feature built into many firewalls that allows the administrator to tweak the tolerance for unanswered login attacks. Reducing this tolerance makes it possible to lessen the likelihood of a successful DoS attack.

If a resource—inbound or outbound—appears to be overused, then the flood guard kicks in. With many Cisco firewalls, to protect subgroups and devices you can configure the same protection you apply at an upper level to be inherited by children as well.

Loop protection is a similar feature that works in layer 2 switching configurations and is intended to prevent broadcast loops. When configuring it in most systems, you can choose to disable broadcast forwarding and protect against duplicate ARP requests (those having the same target protocol address). The Spanning Tree Protocol (STP) is intended to ensure loop-free bridged Ethernet LANs. It operates at the Data Link layer and ensures only one active path exists between two stations.

Preventing Network Bridging

Network bridging occurs when a device has more than one network adapter card installed and the opportunity presents itself for a user on one of the networks to which the device is attached to jump to the other. Although multiple cards have been used in servers (known as multihomed hosts) for years, it is not uncommon today to find multiple cards in laptops (wired and wireless) and the bridging to occur without the user truly understanding what is happening.

To prevent network bridging, you can configure your network such that when bridging is detected, you shut off/disable that jack. You can also create profiles that allow for only one interface.

At a micro level, you can configure workstations to disable unused connections. In Windows 8, for example, you can accomplish this by choosing Control Panel ➢ Network And Internet ➢ Network Sharing Center and then clicking Change Adapter Settings.

It is not uncommon for a network bridge to appear in the Network Sharing Center. If it does appear, you will want to delete it. Windows Internet Connection is often identified as a cause of unintended bridging and should be disabled.
Log Analysis

Log analysis is crucial to identifying problems that occur related to security. As an administrator, you have the ability to turn on logging at many different locations and levels. The next step, however, is the most important—what you do with the log information collected. Far too many administrators turn on logging and then fail to properly (if ever) analyze what they collect because it is a lot of information and a lot of work.

A number of programs are available that can automate the log analysis. One such program is ManageEngine (www.manageengine.com/it-compliance-suite.html).

Not only do you need to collect and analyze the logs, but you also need to store them for the future when you want to compare what is happening now to then (baselining). The logs should be stored in a format that you can quickly access and understand without having to convert them to a document each time you want to look at them.

Trusted OS

A trusted operating system (TOS) is any operating system that meets the government’s requirements for security. The most common set of standards for security is Common Criteria (CC). This document is a joint effort among Canada, France, Germany, the Netherlands, the United Kingdom, and the United States. The standard outlines a comprehensive set of evaluation criteria, broken down into seven Evaluation Assurance Levels (EALs). EAL 1 to EAL 7 are discussed here:

As of this writing, the latest version of the standard is 3.1 Release 4, and it’s available for viewing at www.commoncriteriaportal.org. The website also maintains a registry of products certified by CC.

EAL 1 EAL 1 is primarily used when the user wants assurance that the system will operate correctly but threats to security aren’t viewed as serious.

EAL 2 EAL 2 requires product developers to use good design practices. Security isn’t considered a high priority in EAL 2 certification.

EAL 3 EAL 3 requires conscientious development efforts to provide moderate levels of security.

EAL 4 EAL 4 requires positive security engineering based on good commercial development practices. It is anticipated that EAL 4 will be the common benchmark for commercial systems.

EAL 5 EAL 5 is intended to ensure that security engineering has been implemented in a product from the early design phases. It’s intended for high levels of security assurance. The EAL documentation indicates that special design considerations will most likely be required to achieve this level of certification.
EAL 6  EAL 6 provides high levels of assurance of specialized security engineering. This certification indicates high levels of protection against significant risks. Systems with EAL 6 certification will be highly secure from penetration attackers.

EAL 7  EAL 7 is intended for extremely high levels of security. The certification requires extensive testing, measurement, and complete independent testing of every component.

EAL certification has replaced the Trusted Computer Systems Evaluation Criteria (TCSEC) system for certification, which was popular in the United States. It has also replaced the Information Technology Security Evaluation Criteria (ITSEC), which was popular in Europe. The recommended level of certification for commercial systems is EAL 4.

Currently, only a few operating systems have been approved at the EAL 4 level, and even though an operating system straight out of the box may be, that doesn’t mean your own individual implementation of it is functioning at that level. If your implementation doesn’t use the available security measures, then you’re operating below that level.

As an administrator, you should know and thoroughly understand that just because the operating system you have is capable of being certified at a high level of security doesn’t mean that your implementation is at that level.

**Secure Router Configuration**

One of the most important things you can do to secure your network is to secure the router. Though this is basic common sense, it is too often overlooked in the rush to finish the router configuration and move on to the next job. To configure the router securely, you must do the following:

**Change the default password.** The password for the administrator is set before the router leaves the factory. You have to assume that every intruder wanting unauthorized access to your network knows the default passwords set by the factory. Employ good password principles (alphanumeric, more than eight characters, and so on), and change it to a value that is known only by those who must.

**Walk through the advanced settings.** These settings will differ based on the router manufacturer and type, but they often include settings to block ping requests, perform MAC filtering, and so on. All of these issues are discussed elsewhere in this book, and they need to be applied to the router configuration as well.

**Keep the firmware upgraded.** Router manufacturers often issue patches when problems are discovered. Those patches need to be applied to the router to remove any security gaps that may exist.

Always remember to back up your router configuration before making any significant changes—in particular a firmware upgrade—in order to provide a fallback in case something goes awry.
Cisco routers often use one of two different types of passwords for their accounts: Type 7 and MD5. Type 7 passwords use weak encryption and are considered only slightly above Type 0, which is cleartext. As such, Type 7 passwords are easily decrypted with readily available shareware/freeware and should be avoided. MD5 password encryption uses a one-way hash, and this is configured in IOS (Cisco Internetwork Operating System) using the command enable secret.

**Summary**

The chapter focused on access control and identity management. The key difference between authentication and identification is that authentication means someone has accurate information, whereas identification means that accurate information is proven to be in possession of the correct individual.

The most basic form of authentication is known as single-factor authentication (SFA) because only one set of values is checked. To increase security, it is necessary to use multi-factor authentication, which involves two or more values that are checked.

This chapter examined the various types of authentication services in use, including RADIUS and different variations of TACACS. It also looked at tunneling protocols, smart cards, and other means of access control.

ACLs are being implemented in network devices and systems to enable the control of access to systems and users. ACLs allow individual systems, users, or IP addresses to be ignored.

**Exam Essentials**

**Be able to describe the roles of access control.** The four primary roles are MAC, DAC, and RBAC (both types of RBAC). Mandatory Access Control (MAC) establishes rigid access control methods in the organization. Discretionary Access Control (DAC) allows for flexibility in access control. Role-Based Access Control (RBAC) is based on the role the individual or department has in the organization. In a fourth type, Rule-Based Access Control (RBAC) settings in preconfigured security policies are used to make all decisions.

**Know the characteristics of the connectivity technologies available to you and the security capabilities associated with each.** Remote access, PPP, tunneling protocols, and VPNs are your primary tools. PPTP and L2TP are two of the most common protocols used for tunneling. IPSec, although not a tunneling protocol, provides encryption to tunneling protocols; it’s often used to enhance tunnel security.
Know how ACLs work. Access control lists (ACLs) are used to identify systems and specify which users, protocols, or services are allowed. ACL-based systems can be used to prevent unauthorized users from accessing vulnerable services.

Explain the relative advantages of the technologies available to you for authentication. You have many tools available to you to help establish authentication processes. Some of these tools start with a password and user ID. Others involve physical devices or the physical characteristics of the person who is requesting authentication.

Know how to deal with users having multiple roles. Users who have multiple accounts and/or act in multiple roles, such as an administrator who has an account for administrative purposes and one for performing another role, require attention and education. Users need to understand why they should use the elevated accounts only when necessary and the security risks inherent in operating at those levels.

Understand least privilege. Least privilege states that when assigning permissions, you should give users only the permissions they need to do their work and no more. The biggest benefit to following this policy is the reduction of risk.